Docket No. AT9-99-149

CLAIMS:

What is claimed is:

Sub A1

1. A method in a data processing system for monitoring a plurality of related threads, the method comprising the data processing system implemented steps of:

polling the plurality of related threads for status information;

responsive to receiving the status information, determining whether a thread within a plurality of related threads is active; and

responsive to an absence of a determination that a thread within the plurality of related threads is active, initiating cleanup processes for the thread based on the status information.

2. The method of claim 1 further comprising:

responsive to receiving the status information, storing the status information.

Sub B1

3. The method of claim 1, wherein the polling, determining, and initiating steps are performed by a single thread.

Sub A2

4. The method of claim 1, wherein the single thread is part of a first class

5. The method of claim 1, wherein the initiating step comprises:

identifying active threads within the plurality of related threads;

identifying inactive threads within the plurality of related threads; and

terminating inactive threads.

6. The method of claim 1, wherein the step of terminating inactive threads includes:

Sub B1

resetting resources allocated to an identified inactive thread such that the resources are reallocatable.

7. The method of claim 1, wherein the plurality of related threads is a plurality of printer threads.

8. The method of claim 1, wherein the plurality of related threads is a plurality of video threads.

9. The method of claim 1, wherein the method is implemented in a virtual machine.

10. The method of claim 9, wherein the virtual machine is a Java virtual machine.

Sub A3

11. A method in a data processing system for monitoring a plurality of related threads, the method comprising the data processing system implemented steps of:

polling the plurality of related threads for status information;

responsive to receiving the status information, determining whether an error has occurred in a thread within a plurality of related threads is active; and

responsive to an occurrence of inactivity in a thread within the plurality of related threads in which the inactivity is due to an event, initiating cleanup processes based on the status information.

A3 end

12. The method of claim 11, wherein the event is a period of time.

Sub B1

13. The method of claim 11, wherein the event is an error.

Sub A4

14. A data processing system for monitoring a plurality of related threads, the data processing system comprising:

polling means for polling the plurality of related threads for status information;

determining means, responsive to receiving the status information, for determining whether a thread within a plurality of related threads is active; and

initiating means, responsive to an absence of a determination that a thread within the plurality of related threads is active, for initiating cleanup processes for the thread based on the status information.

15. The data processing system of claim 14 further comprising:

storing means, responsive to receiving the status information, for storing the status information.

Sub B1

16. The data processing system of claim 14, wherein the polling, determining, and initiating means are preformed by a single thread.

Sub A5

17. The data processing system of claim 14, wherein the single thread is part of a first class

18. The data processing system of claim 14, wherein the initiating means comprises:

first identifying means for identifying active threads within the plurality of related threads;

second identifying means for identifying inactive threads within the plurality of related threads; and

terminating means for terminating inactive threads.

Sub B1

19. The data processing system of claim 14, wherein the means of terminating inactive threads includes:

resetting means for resetting resources allocated to an identified inactive thread such that the resources are reallocatable.

20. The data processing system of claim 14, wherein the plurality of related threads is a plurality of printer threads.

21. The data processing system of claim 14, wherein the plurality of related threads is a plurality of video threads.

22. The data processing system of claim 14, wherein the data processing system is implemented in a virtual machine.

23. The data processing system of claim 22, wherein the

virtual machine is a Java virtual machine.

Sub A6

24. A data processing system for monitoring a plurality of related threads, the data processing system comprising:

polling means for polling the plurality of related threads for status information;

determining means, responsive to receiving the status information, for determining whether an error has occurred in a thread within a plurality of related threads is active; and

initiating means, responsive to an occurrence of inactivity in a thread within the plurality of related threads in which the inactivity is due to an event, for initiating cleanup processes based on the status information.

25. The data processing system of claim 24, wherein the event is a period of time.

Sub B1

~~26. The data processing system of claim 24, wherein the event is an error.~~

Sub A7

27. A computer program product in a computer readable medium for monitoring a plurality of related threads, the computer program product comprising:

first instructions for polling the plurality of related threads for status information;

second instructions for responsive to receiving the status information, determining whether a thread within a plurality of related threads is active; and third instructions for responsive to an absence of a determination that a thread within the plurality of related threads is active, initiating cleanup processes for the thread based on the status information.

28. A computer program product in a computer readable medium for monitoring a plurality of related threads, the computer program product comprising:

first instructions for polling the plurality of related threads for status information;

second instructions, responsive to receiving the status information, for determining whether an error has occurred in a thread within a plurality of related threads is active; and

third instructions, responsive to an occurrence of inactivity in a thread within the plurality of related threads in which the inactivity is due to an event, for initiating cleanup processes based on the status information.

A7 end